

Ing1 - Examen de Rattrapage d'Architecture des Ordinateurs

Réponses

Exercice 1 :

1. une adresse sur 32 bits permet d'adresser au maximum 2^{32} mots de 2 octet soit $2^{33} \text{ octets} = 2^{23} \text{ Ko} = 2^{13} \text{ Mo} = 2^3 \text{ Go} = 8 \text{ Go}$
2. plage d'adresses :
 - 0 à $2^{32} - 1$
 - $00000000000000000000000000000000_{(2)}$ à $11111111111111111111111111111111_{(2)}$ (32 chiffres)
 - $00000000_{(16)}$ à $FFFFFFFF_{(16)}$ (8 chiffres)
3. registre d'adresse MAR : 32 bits ; registre de donnée MDR : 16 bits.
4. Nombre d'accès mémoire :
 - LOAD R1, 127 ; 3 pour charger l'instruction, 0 pour la donnée contenue dans l'instruction.
 - LOAD R1, (127) ; 4 : 3 pour charger l'instruction et 1 pour charger la donnée (1 mot) à l'adresse 127
 - LOAD R1, ((127)) ; 6 : 3 pour charger l'instruction, 2 pour charger l'adresse de la donnée contenue dans la mémoire à l'adresse 127 (une adresse sur 32 bits tient sur 2 mots mémoire) et 1 pour charger la donnée (1 mot) contenue à cette adresse.

Exercice 2 :

La mémoire fait 256 octets ($2^8 * 1$ octet), 1 boîtier 64 octets ($2^7 * 1/2$ octet).

- on utilise 4 boîtiers mémoires (2 rangées de 2 boîtiers)
- les 4 bits de poids forts du bus de données sont connectés au bus de donnée du 1er boîtier de chaque rangée.
- les 4 bits de poids faibles du bus de données sont connectés au bus de donnée du 2ème boîtier de chaque rangée
- les 7 bits de poids faibles de l'adresse (6 à 0) sont connectés sur les broches d'adressage des 4 boîtiers
- le bit de poids fort de l'adresse passent à travers un décodeur $1 \rightarrow 2$ qui activera une seule rangée de boîtiers à la fois
- le décodeur est lui même activé par le signal d'activation de la mémoire

Exercice 3 :

1. le codage des nombres entiers relatifs en complément à 2 sur 6 bits permet de représenter les nombres de -32 (-2^5) à 31 ($2^5 - 1$).
2. il y a dépassement de capacité lors d'une addition de deux entiers relatifs en complément à 2 si et seulement si les 2 dernières retenues (sur les poids forts) sont différentes.

Codage des nombres en binaire complément à 2 sur 6 bits :

- $5_{(10)} = 101_{(2)} \rightarrow C2 : 000101$
- $-3_{(10)} : 3_{(10)} = 11_{(2)} \rightarrow 000011 \rightarrow 111100 \rightarrow C2 : 111101$
- $-16_{(10)} : 16_{(10)} = 10000_{(2)} \rightarrow 010000 \rightarrow 101111 \rightarrow C2 : 110000$
- $-18_{(10)} : 18_{(10)} = 10010_{(2)} \rightarrow 010010 \rightarrow 101101 \rightarrow C2 : 101110$

$$5 + (-3) : \begin{array}{r} 111101 \\ 000101 \\ +111101 \\ \hline 1000010 \end{array}$$

Sur 6 bits, on obtient bien le bon résultat $000010_{(C2)} = 2_{(10)}$. Les 2 dernières retenues sont égales à 1 pas de dépassement de capacité.

$$(-16) + (-18) : \begin{array}{r} 100000 \\ 110000 \\ +101110 \\ \hline 1011110 \end{array}$$

Sur 6 bits, il nous reste $011110_{(C2)}$ qui représente un nombre positif. Le résultat est donc faux. En effet, les 2 dernières retenues sont différentes (1 et 0) : il y a dépassement de capacité.

Exercice 4 :

1. – 4 bits de poids faible : numéro du mot dans le bloc
- 20 bits de poids forts : numéro de bloc dans la mémoire
- 8 bits du poids 11 à 4 : numéro d'ensemble, correspondant au modulo 2^8 du nombre d'ensembles (en fait on peut prendre n'importe quel sous-ensemble fixe de 8 bits parmi les 20 bits de numéro de bloc)
- 12 bits du poids 23 à 12 : TAG de reste d'ensemble (ceux qu'on n'a pas pris pour le numéro d'ensemble dans les 20 bits de numéro de bloc).

	num bloc	LRU	num bloc	LRU	num bloc	LRU	num bloc	LRU	num bloc	LRU	num bloc	LRU
	12	0	-	1	-	2	-	3	-	4	-	5
	504	3	-	4	504	0	-	1	-	2	-	3
2.	496	5	-	6	-	-	-	7	*260	0	-	1
	48	4	-	5	-	-	-	6	-	7	*496	0
	128	7	*260	0	-	1	-	2	-	3	-	4
	44	2	-	3	-	4	-	5	-	6	-	7
	688	1	-	2	-	3	-	4	-	5	-	6
	224	6	-	7	-	-	224	0	-	1	-	2

- : aucun changement

* : défaut de cache

Exercice 5 :

$$-0,4FAB0D_{(16)} = -0,010011111010101100001101_{(2)} = (-1)^1 * (1,001111101010101100001101_{(2)}) * 2^{-2}$$

$$-S = 1$$

$$-E = -2 + 127 = 125 = 01111101_{(2)}$$

$$-M = 0011111010101011000011010 \text{ (on complète par un 0)}$$

D'où le code IEEE754 simple précision de $-0,4FAB0D_{(16)}$:

1 01111101 00 1111 1010 1011 0000 1101 0

Exercice 6 :

LD IX, 100h

LD A, (IX)

; addition des octets de poids faibles

ADD A, (IX+8)

LD (IX) A

LD B,3

; compteur de boucle pour les 3 octets suivants

bcle : INCR IX

; on passe à l'adresse de l'octet suivant

LD A,(IX)

; addition d'un octet avec retenue du précédent

ADC A,(IX+8)

LD (IX),A

DJNZ bcle

; B=B-1 et rebouclage jusqu'à B=0